

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**  
**BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of: § Group Art Unit: 2166  
§  
Ajay Kumar, et al. § Examiner: Hwang, Joon H.  
§  
§ Atty. Dkt. No.: 5681-11800  
§ P7519  
Serial No. 10/087,197 §  
§  
§  
Filed: March 1, 2002 §  
§  
For: Incremental Saves for §  
Efficient Distributed State §  
Stores §  
§  
§  
§  
§

**REPLY BRIEF**

**Mail Stop Appeal Brief - Patents**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir/Madam:

This brief is in reply to the Examiner's Answer dated February 26, 2007. Appellants respectfully request that this Reply Brief be entered pursuant to 37 C.F.R. § 41.41 and considered by the Board of Patent Appeals and Interferences.

## **REPLY TO EXAMINER'S ANSWER**

### **First Ground of Rejection**

Claims 1, 3-5, 8, 9, 11, 12, 15, 17 and 18 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Montero in view of Bauer. Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

#### **Claims 1, 8, 9, 11, 15 and 17**

Appellants have argued, regarding claim 1, that Montero fails to teach or suggest a first application server of the plurality of application servers, comprising a client state of the session data accessible to processes executing within the application server, **wherein the first application server is configured to track accesses of the individual attributes of the client state, wherein to track accesses of the individual attributes of the client state, the first application server is configured to store information identifying the accessed individual attributes.**

The Examiner seems to assert that Montero suggests a need to store information identifying the accessed individual attributes because Montero teaches that “every change to the session data is written to the common database after each request or every time a change occurs.” The Examiner’s characterization of Montero’s teachings is incorrect. In fact, Montero actually teaches away from purposefully updating the common session database after each request. Since Montero explicitly does not teach purposefully updating the common session database after each request, there would be no reason in Montero’s system to store information identifying the accessed individual attributes. Thus, Montero clearly cannot be said to teach or suggest this feature. The Examiner asserts, in the Examiner’s Answer, that Montero teaches that “every change to the session data is written to the common database after each request or every time a change occurs”, citing various portions of Montero including: paragraph 20 (lines 3-5), paragraph 44

(lines 2- 5), paragraph 45 (lines 3-4), paragraph 49, paragraph 50 (lines 18-21) paragraph 53 (lines 24-30). However, the Examiner has mischaracterized the teachings and language of Montero. Specifically, the Examiner has taken individual sentences of Montero out of context. For example, while paragraph 20, lines 3-5 does state, “the server updates the http session data in its local memory and also writes data to the database after each request”, this paragraph is part of Montero’s background section **and does not describe Montero’s system that the Examiner otherwise relies upon.** Moreover, Montero states that some systems update session data after each request as an introduction to stating, “Writing to the database is a relative[ly] expensive process in terms of consumption of processing power and time. Accordingly, it is desirable to reduce the number of writes to an http session database in order to conserve system resources.” Thus, Montero mentions updating session data after each request as a lead-in to discussing the disadvantages to doing that very thing. Therefore, Montero actually teaches away from updating session data after each request.

Similarly, the Examiner has misinterpreted Montero’s teachings at paragraph 44, lines 2-5. The Examiner contends that Montero is describing updating the common session database after each request. However, the Examiner fails to consider that at paragraph 44, lines 2-5 state that a “write to the common session database should be performed *at the end of the predetermined interval* only if the session data has been modified *since the last write to the session database.*” Thus, the Examiner has clearly ignored the full and complete teaching of Montero.

Moreover, paragraph 44 of Montero is not describing updating session data *after each request*, as the Examiner contends, but instead, Montero is talking about updating the common session database from a local session database periodically (e.g., “at the end of the predetermined interval”). Montero makes this perfectly clear at the end of paragraph 44 by stating, “It will be understood by persons of skill in these arts that the modification to the session attributes may have occurred over multiple http requests.”

The Examiner’s other citations also either teach away from updating the common

session database after each request or otherwise support Appellants' argument. For instance, the Examiner's cited passage at paragraph 50 also supports Appellants' argument. Montero states, "writing of the initial session data to the session database may be deferred just as all other writes to the database will be deferred." This statement by Montero clearly fails to teach updating the session database after each request.

At paragraph 45, Montero further reinforces the fact that his system is not designed, nor configured, to update the common session database after each request. Montero states that only if the interval between requests was longer than the predetermined interval would the common session database be updated after each request. Montero, when taken in its entirety, even if combined with Bauer, clearly teaches away from *purposefully* updating the common session database after each request, even while his system will not prevent an update after each request if the http requests occur seldom enough.

Additionally, the Examiner has overlooked or ignored the clear and direct statements by Montero. For example, the Examiner is clearly overlooking the statement by Montero that, "In accordance with the invention, instead of updating the session data in the database after every request or every attribute change, each of the servers maintains a fully current copy of the http session data in its local RAM, but writes a copy of the session data to the central database only at specified intervals" (Montero, paragraph 39).

Furthermore, Appellants are not arguing that Montero's system cannot update a common session database after each request. Appellants are arguing that Montero in view of Bauer does not teach or suggest an application server configured to store information identifying accessed individual attributes to track accesses of individual attributes. Appellants' argument is, in part, that since, as shown above, Montero does not teach purposefully updating the common session database after each request, there is no reason in Montero's system to store information identifying the accessed individual attributes. To function as intended Montero's system does not need to store information identifying the accessed individual attributes. Montero clearly cannot be said to teach

storing information identifying the accessed individual attributes, and the prior art does not suggest any reason to modify Montero's system to store such information.

Bauer teaches that each of the clients (which are not application servers) accesses its own client database, which may later be synchronized with the single central database by the database synchronizer. Appellants note that a database as described in Bauer refers to a collection of data that is manipulated by clients. More specifically, Bauer's data is described as tabular data for order information, delivery status, or field service information (1:4-14), which may be manipulated by clients in a database. The order information, delivery status, or field service information stored in the database in Bauer is not *session data*. Session data is a well-understood concept in the art of application servers, and the data stored in the databases in Bauer is clearly not described as session data. Bauer does not pertain to the field of session data as used by application servers. Thus, the Examiner's reliance on Bauer to teach an application server configured to store information identifying the accessed individual attributes of session data is clearly improper.

Since Bauer has nothing to do with application servers or session data, there would be no reason to apply its teachings to Montero, especially in light of the fact that Montero specifically teaches writing a copy of the session data to the central database "only at specified intervals" "instead of updating the session data in the database after every request or every attribute change". Thus, for at least the reasons provided above, Montero in view of Bauer fails to teach or suggest *wherein to track accesses of the individual attributes of the client state, the first application server is configured to store information identifying the accessed individual attributes*.

The Examiner also asserts, in the Examiner's Answer, that Appellants' have mischaracterized Montero's teaching regarding updating the common session database. The Examiner contends that Montero's common session database is updated only if the session data has been modified and that "if the session data is not changed, then the session data would not be written to the common session database." The Examiner, then

concludes, “[t]herefore, the applicants’ ‘Counting the number of times session data is updated and then writing the entire copy of the session data...’ is mischaracterized, since Montero would write only the changed copy of the session data as discussed above”. However, the fact that Montero would only update the common session database if the session data changed does not imply the method or details of updating the common session database (e.g., whether the entry data is copied or whether only the changed data is copied).

Appellants submit that the Examiner has overlooked the clear and plain language of Montero’s teachings. For example, Montero states, “[t]he servers write a copy of the session data to a common database” (paragraph 26) and, as noted above, “each of the servers maintains a fully current copy of the http session data in its local RAM, but writes a copy of the session data to the central database only at specified intervals” (paragraph 39). Thus, Montero clearly states that the servers “write a copy of the session data” to the central database. Montero does not teach, as the Examiner contends, writing only the changed data or attributes. The Examiner is clearly speculating in hindsight about the details of Montero’s system while overlooking the direct statements by Montero.

Similarly, the Examiner contends, (page 11 of Answer) that Montero’s system “would have to monitor and/or identify accesses and any changes to session data in the application server in order to trigger the synchronization.” However, the Examiner’s interpretation is again based on hindsight speculation and not the actual teachings of the reference. The Examiner’s argument is that since Montero defers updating the common session database (or may update the common session database based on the number of requests or changes to the session), Montero’s system would “have to ... identify accesses and any changes to session data.” The Examiner logic is incorrect. Montero, as evidenced by the Examiner’s citations, teaches updating the common session database based on a predetermined time interval, the number of changes or requests since the last time the common session database was updated, or a combination of those criteria. **None of the criteria taught by Montero requires, or would even benefit from, storing information identifying the individual access attributes.** Instead, Montero’s system

only needs to keep track of how long it has been since the common session database was last updated and/or how many requests or updates have occurred since the common session database was last updated.

Similarly, the Examiner asserts that Montero’s “application server would track/monitor accesses to the session data (i.e., session attributes) in order to keep track of the number of requests.” Again, the Examiner is incorrect. Montero’s application server would only have to keep a count of the number of requests. There is no reason or benefit to identifying accessed individual attributes to keep track of a number of requests. A count is all that is needed. Montero repeatedly describes keeping track of the number of changes and the number of requests but does not suggest, even if combined with Bauer, storing information identifying the accessed individual attributes (See, e.g., paragraphs 26, 40, 45, 49).

In the Examiner’s Answer, the Examiner repeats his earlier contention that Appellants’ previous arguments were against the cited referenced individually. To the contrary, Appellants’ arguments above show that Montero and Bauer, whether considered alone or in combination, fail to teach or suggest that the first application server is configured to track accesses of the individual attributes of the client state, wherein to track accesses of the individual attributes of the client state, the first application server is configured to store information identifying the accessed individual attributes, as recited in claim 1. Moreover, any statement regarding an individual reference is to show that the Examiner’s reliance on the respective reference is misplaced.

Additionally, Appellants have argued that the Examiner has not provided a proper motivation for modifying Montero in view of Bauer. Appellants have further argued that Bauer and Montero teach away from each other. Bauer’s system seeks to minimize delays in synchronization of modified client data with a database storing data that is not fully current while Montero purposefully introduces delays by only updating at specified time intervals or after a certain number of updates. Thus, Montero reduces the number of writes to the database by purposefully introducing delays in the form of

periodic writes, and maintains a fully current copy of the session data while Bauer, on the other hand, minimizes delay in synchronization of data between the client and the database, where neither the client nor the database maintain a fully current copy of the tabular data. Therefore, Montero and Bauer clearly teach away from their combination. It is improper to combine references where the references teach away from their combination. See Appellants' Appeal Brief, pp. 14-16.

In the Advisory Action, the Examiner argues that "it would have been obvious ... to utilize the teaching of Bauer to the system of Montero in order to minimize the cost of synchronization." Specifically, the Examiner argues that Montero discloses synchronization of session data in a common session database with session data in an application server and Bauer discloses synchronization of a target database of a server with a source database of a client of the server. The Examiner also asserts that Bauer teaches storing information identifying operations, such as updates, insertion and deletion of data. The Examiner then concludes, "based on Montero in view of Bauer, it would have been obvious ... to utilize the teaching of Bauer to the system of Montero in order to minimize the cost of synchronization."

However, as noted above, Montero and Bauer clearly teach away from their combination. Since Montero specifically teaches away from updating the session data in the database after every request or every attribute change, there would be no reason *to store information identifying the accessed individual attributes* to track individual attribute accesses in Montero. Thus, it would not make sense to modify Montero according to the teachings of another reference to store information identifying the accessed individual attributes to track individual attribute accesses.

Furthermore, even were Montero and Bauer combinable, which Appellants argue they are not, the suggested combination would not provide all the features recited in claim 1, as described above. As noted in the M.P.E.P 2143.03, "[t]o establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art." *In re Royka*, 490 F.2d 981, 180 USPQ 580 (CCPA 1974). As shown

above, neither reference, whether considered alone or in combination, teaches or suggests a first application server of the plurality of application servers, comprising a client state of the session data accessible to processes executing within the application server, wherein the first application server is configured to track accesses of the individual attributes of the client state, wherein to track accesses of the individual attributes of the client state, the first application server is configured to store information identifying the accessed individual attributes.

### **Claim 3**

Regarding claim 3, Appellants have argued that Montero in view of Bauer fails to teach or suggest that *the first application server is further configured to track mutable individual attributes and not track immutable individual attributes.*

The Examiner relies on Bauer, citing col. 3, lines 53-60, col. 7, lines 43-36 and col. 28, lines 28-52, where Bauer describes that a developer may specify which fields are not modifiable at a client to enable Bauer's database synchronizer to "omit such fields in the before-image table as long as the fields are also not used to uniquely identify a given row." In the Examiner's Answer, the Examiner asserts that Bauer's database synchronizer uses "a smart differencing technique which enables the database synchronizer to identify data items as being mutable and immutable and to track mutable data items and not to track immutable data items." However, as noted above regarding the rejection of claim 1, Since Montero specifically teaches away from updating the session data in the database after every request or every attribute change, there would be no reason to track individual attribute accesses in Montero. Similarly, there would be no reason to track mutable individual attributes and not track immutable individual attributes in Montero.

Moreover, even were Montero and Bauer combinable, which Appellants argue they are not, as argued above, the suggested combination would not provide all the features recited in claim 3, as described above.

#### Claim 4

Regarding claim 4, Appellants have argued that Montero in view of Bauer fails to teach or suggest that to synchronize the primary state with the client state, **the distributed store is further configured to synchronize only mutable individual attributes**. The Examiner, in the Examiner's Answer, relies on Bauer, referring to Bauer's database synchronizer and "smart differencing". However, as noted above regarding the rejection of claim 1, since Montero specifically teaches away from updating the session data in the database after every request or every attribute change, there would be no reason to synchronize individual attributes in Montero. Similarly, there would be no reason to synchronize only mutable individual attributes in Montero. Moreover, it would not make sense to modify Montero according to the teachings of another reference to synchronize only mutable individual attributes. Furthermore, even were Montero and Bauer combinable, which Appellants argue they are not, as argued above, the suggested combination would not provide all the features recited in claim 4, as described above.

Thus, for at least the reasons above, the rejection of claim 4 is not supported by the cited art and removal thereof is respectfully requested.

#### Claims 5, 12 and 18

Regarding claim 5, the Examiner asserts that "Montero teaches the distributed store is configured to update the primary state with the subset of the accessed attributes that have been modified for synchronizing the primary state with the client state" in paragraph [0026]. However, paragraph [0026] summarizes Montero's system including a description of how application servers write copies of session data to common shared databases at designated, periodic times, or after a specified number of changes to session data have been made. The cited art does not mention anything about *the distributed store being configured to update the primary state with the subset of the accessed individual attributes that have been modified*. Contrary to the Examiner's assertion, Montero

clearly fails to teach or suggest this claim limitation.

In the Examiner's Answer, the Examiner argues that Montero saving a last access time indicating the last time an HTTP request was received teaches a "subset of the accessed individual attributes that have been modified." However, The Examiner interpretation of Montero's teaching is incorrect. Montero teaches saving the last access time in order to determine whether there were any updates to a local session database since the last time the common session database was updated. Saving a last access time does not teach anything about updating with a subset of the accessed attributes that have been modified. Instead, as taught by Montero, the last access time will change "every time a request is received" (Montero, paragraph 38). Thus, the last access time indicates when the last request was received. Moreover, as noted above regarding claim 1, Montero teaches copying the entire session data to the common session database. Thus, according to Montero's own teaching, when synchronizing, if the last access time is later than the last time the common session database was updated, the session data is copied to the common session database. The Examiner's contention that Montero's use of a last access time teaches the limitation of claim 5 is clearly incorrect.

Furthermore, the Examiner admits that Montero does not disclose performing a comparison of the tracked accessed attributes and a benchmark of the session data comprising a previous version of the one or more attributes. Thus, the Examiner relies on Bauer to disclose this claim limitation. However, as noted above regarding the rejection of claim 1, since Montero specifically teaches away from updating the session data in the database after every request or every attribute change, there would be no reason to perform a comparison of the tracked accessed individual attributes and a benchmark of the session data comprising a previous version of the one or more individual attributes to determine a subset of the tracked accessed individual attributes that are modified in respect to the benchmark of the session data in Montero. Moreover, it would not make sense to modify Montero according to the teachings of another reference to determine a subset of the tracked accessed individual attributes. Furthermore, even were Montero and Bauer combinable, which Appellants argue they are not, as argued above, the suggested

combination would not provide all the features recited in claim 5, as described above.

### **Second Ground of Rejection**

Claims 6, 13 and 19 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Montero in view of Bauer and further in view of Morris. Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

#### **Claims 6, 13, 19**

Regarding claim 6, Appellants have argued that Montero in view of Bauer in further view of Morris fails to disclose ***wherein, in said comparison, the first application server is configured to perform a binary comparison of the tracked accessed individual attributes and the benchmark of the session data to determine a subset of the tracked accessed individual attributes that are modified in respect to the benchmark of the session data.*** The Examiner relies on Morris to teach this claim limitation. In the Examiner's Answer, the Examiner cites the abstract and column 11, line 47 through column 12, line 13. However, as noted in Appellants' Appeal Brief, these citations describe a method of comparing a compressed file to a base file by compressing the base file and performing a binary comparison of the compressed file and the compressed base file. Morris fails to disclose ***wherein, in said comparison, the first application server is configured to perform a binary comparison of the tracked accessed individual attributes and the benchmark of the session data to determine a subset of the tracked accessed individual attributes that are modified in respect to the benchmark of the session data.***

Appellants have also argued that no teaching, motivation, or suggestion exists to combine the teachings of Morris with the teachings of Montero. In the Examiner's Answer, the Examiner asserts “[t]herefore, based on Montero in view of Bauer, and further in view of Morris, it would have been obvious to one having ordinary skill in the

art at the time the invention was made to utilize the teachings of Morris to the system of Montero in order to reduce the cost of synchronization" (emphasis added). However, as noted above regarding the rejection of claim 1, Montero and Bauer teach away from their combination. Moreover, even if Montero and Bauer were combinable, which Appellants argue they are not, one with ordinary skill in the art would simply use the teachings of Bauer to determine differences between two versions of data.

Additionally, as noted above regarding the rejection of claim 1, Montero specifically teaches away from updating the session data in the database after every request or every attribute change; therefore, there would be no reason to perform a binary comparison of the tracked accessed individual attributes and the benchmark of the session data to determine a subset of the tracked accessed individual attributes that are modified in respect to the benchmark of the session data in Montero. Moreover, it would not make sense to modify Montero according to the teachings of another reference to determine a subset of the tracked accessed individual attributes. Additionally, the prior art references lack a teaching or suggestion to combine the teachings of Montero and the teachings of Morris.

### **Third Ground of Rejection**

Claims 7, 14 and 20 stand finally rejected under 35 U.S.C. § 103(a) as being unpatentable over Montero in view of Bauer and further in view of Lin. Appellants traverse this rejection for at least the following reasons. Different groups of claims are addressed under their respective subheadings.

Regarding claim 7, Appellants have argued that Montero in view of Bauer in further view of Lin fails to disclose *wherein, in said comparison, the first application server is configured to perform an object graph comparison of the tracked accessed individual attributes and the benchmark of the session data to determine a subset of the tracked accessed individual attributes that are modified in respect to the benchmark of the session data.* The Examiner, in the Examiner's Answer, relies on Lin, citing the

abstract; column 7, line 40 through column 8, line 14; and Figure 5 of Lin. However, these citations do not disclose *wherein, in said comparison, the first application server is configured to perform an object graph comparison of the tracked accessed individual attributes and the benchmark of the session data to determine a subset of the tracked accessed individual attributes that are modified in respect to the benchmark of the session data*. In contrast, The cited art discloses a method of comparing multimedia content to produce a similarity score. As described above, session data is a well-understood concept in the art of application servers, and the data compared in Lin is clearly not described as session data. Lin does not pertain to the field of session data as used by application servers. Furthermore, Lin fails to teach or suggest the comparison of **tracked accessed individual attributes** and **benchmark session data**. Thus, the Examiner's reliance on Lin to teach an application server that is configured to perform an object graph comparison of the tracked accessed individual attributes and the benchmark of the session data to determine a subset of the tracked accessed individual attributes is clearly improper.

In the Examiner's Answer, the Examiner asserts “[i]herefore, based on Montero in view of Bauer, and further in view of Lin, it would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teachings of Lin to the system of Montero **in order to determine differences of two version of data**, thereby performing an effective database synchronization” (emphasis added). Appellants respectfully submit that no teaching, motivation, or suggestion exists to combine the teachings of Lin with the teachings of Montero. Clearly, one with ordinary skill in the art, based on Montero in view of Bauer and further in view of Lin, would not be motivated to combine the teachings of Lin and Montero to “determine differences of two versions of data.” Even if Montero and Bauer were combinable, which Appellants argue they are not, as argued above, one with ordinary skill in the art would simply use the teachings of Bauer to “determine differences of two versions of data”.

However, as noted above regarding the rejection of claim 1, Montero specifically teaches away from updating the session data in the database after every request or every

attribute change; therefore, there would be no reason to perform an object graph comparison of the tracked accessed individual attributes and the benchmark of the session data to determine a subset of the tracked accessed individual attributes that are modified in respect to the benchmark of the session data in Montero. Moreover, it would not make sense to modify Montero according to the teachings of another reference to determine a subset of the tracked accessed individual attributes. Additionally, the prior art references lack a teaching or suggestion to combine the teachings of Montero and the teachings of Lin.

## CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of the pending claims was erroneous, and reversal of his decision is respectfully requested.

The Commissioner is authorized to charge any fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-11800/RCK. This Reply Brief is submitted with a return receipt postcard.

Respectfully submitted,

/Robert C. Kowert/  
Robert C. Kowert, Reg. #39,255  
Attorney for Appellants

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.  
P.O. Box 398  
Austin, TX 78767-0398  
(512) 853-8850

Date: April 25, 2007